特别提示：本代码为所有顶点的数据类型由整型改为双精度后代码，这样可避免因类型转换引起精度损失。其中被替代的代码仍保留，但已被注释掉。
功能：消去用一个数组表示的节点矢量U中下标为s的节点u，num次。
输入参数：n-控制顶点数减1，k-次数，U-节点矢量，(Pwx,Pwy)-控制顶点，
          u-待消去节点值，s-待消去节点下标，r-待消去节点重复度，
          num-预设消去次数。
输出参数：t-实际消去次数，U-新节点矢量，(Pwx,Pwy)-新控制顶点。

```
void RemoveCuvKnot(int n,int k,CArray<double,double> &U,CArray<double,double> &Pwx,CArray<double,double>
&Pwy,double &u,int &s,int &r,int &num,int &l)
{   /* 消去用一个数组表示的节点矢量U中下标为s的节点u，num次。*/
        /* 输入：n-控制顶点数减1，k-次数，U-节点矢量，Pwx,Pwy-控制顶点，u-待消去节点值，s-待消去节点下
标，r-待消去节点重复度，num-预设消去次数。*/
        /* 输出：l-实际消去次数， U & Pw-新节点矢量和新控制顶点*/
        Pwx.SetSize(n+1);   Pwy.SetSize(n+1);
        int m=n+k+1;
        U.SetSize(m+1);
        CArray<double,double> tempx,tempy;
        tempx.SetSize(2*k+1);   tempy.SetSize(2*k+1);
        int ord=k+1;
        int fout=(2*s-r-k)/2;      /* 首先删除顶点的下标 */
        int last=s-r;
        int first=s-k;
        int i,j,ii,jj;
        double alfi,alfj;
        for(l=0;  l<num; l++)
        {                          /* 这是(8.7)式循环 */
                int off=first-1;       /* 在temp和pw间的下标差 */
                tempx[0]=Pwx[off];  tempy[0]=Pwy[off];
                tempx[last+1-off]=Pwx[last+1];  tempy[last+1-off]=Pwy[last+1];
                i=first;  j=last;
                ii=1;     jj=last-off;
                bool remflag=false;
                while (j-i>l)
                {                      /* 计算一次消去所得新控制顶点 */
                        alfi=(u-U[i])/(U[i+ord+l]-U[i]);
                        alfj=(u-U[j-l])/(U[j+ord]-U[j-l]);
                        tempx[ii]=(Pwx[i]-(1.0-alfi)*tempx[ii-1])/alfi;
tempy[ii]=(Pwy[i]-(1.0-alfi)*tempy[ii-1])/alfi;
                        tempx[jj]=(Pwx[j]-alfj*tempx[jj+1])/(1.0-alfj);
tempy[jj]=(Pwy[j]-alfj*tempy[jj+1])/(1.0-alfj);
                        i=i+1;   ii=ii+1;
                        j=j-1;   jj=jj-1;
                }                      /* while循环结束 */
                if(j-i<l)              /* 检查节点可否消去 */
                {
                        if(Distance(tempx[ii-1],tempy[ii-1],tempx[jj+1],tempy[jj+1]) <= TOL)
remflag=true;
                }
                else
                {
                        alfi=(u-U[i])/(U[i+ord+l]-U[i]);
                        if(Distance(Pwx[i],Pwy[i],alfi*tempx[ii+l+1]+(1.0-alfi)*tempx[ii-1],
                                              alfi*tempy[ii+l+1]+(1.0-alfi)*tempy[ii-1]) <= TOL)
remflag=true;
                }
                if(remflag==false)     /* 不能再消去节点 */
                        break;             /* 跳出for循环 */
                else
                {                      /* 成功消去，保存新控制顶点 */
                        i=first;  j=last;
                        while (j-i>l)
                        {
                                Pwx[i]=tempx[i-off];    Pwy[i]=tempy[i-off];
                                Pwx[j]=tempx[j-off];    Pwy[j]=tempy[j-off];
                                i=i+1;   j=j-1;
                        }
                }
                first=first-1;   last=last+1;
        }                           /* 结束for循环 */
        if(l==0) return;
        int kk;
        for(kk=s+1; kk<=m; kk++)
                U[kk-l]=U[kk];          /* 节点移位 */
        j=fout;  i=j;
        for(kk=1; kk<l; kk++)       /* 重写pj到pi */
```

```
                if(fmod(double(kk),2.)==1.) i=i+1; /* 取kk除以2的余数 */
                else j=j-1;
        for(kk=i+1; kk<=n; kk++)      /* 顶点移位 */
        {
                Pwx[j]=Pwx[kk];   Pwy[j]=Pwy[kk];  j=j+1;
        }
        return;
}

void CN01Doc::Remove3DCuvKnot(int n,int k,CArray<double,double> &U,CArray<double,double>
&Pwx,CArray<double,double> &Pwy,CArray<double,double> &Pwz,double &u,int &s,int &r,int &num,int &l)
{   /* 消去用一个数组表示的节点矢量U中下标为s的节点u，num次。*/
        /* 输入：n-控制顶点数减1，k-次数，U-节点矢量, Pwx,Pwy,Pwz-控制顶点，u-待消去节点值，s-待消去节点
下标，r-待消去节点重复度, num-预设消去次数。*/
        /* 输出：l-实际消去次数， U & Pw-新节点和新控制顶点*/
        Pwx.SetSize(n+1);   Pwy.SetSize(n+1);   Pwz.SetSize(n+1);
        int m=n+k+1;
        U.SetSize(m+1);
        CArray<double,double> tempx,tempy,tempz;
        tempx.SetSize(2*k+1);   tempy.SetSize(2*k+1);   tempz.SetSize(2*k+1);
        int ord=k+1;
        int fout=(2*s-r-k)/2;      /* 首先删除顶点的下标 */
        int last=s-r;
        int first=s-k;
        int i,j,ii,jj;
        double alfi,alfj;
        for(l=0;  l<num; l++)
        {                          /* 这是(8.7)式循环 */
                int off=first-1;       /* 在temp和pw间的下标差 */
                tempx[0]=Pwx[off];  tempy[0]=Pwy[off];  tempz[0]=Pwy[off];
                tempx[last+1-off]=Pwx[last+1];  tempy[last+1-off]=Pwy[last+1];
tempz[last+1-off]=Pwz[last+1];
                i=first;  j=last;
                ii=1;     jj=last-off;
                bool remflag=false;
                while (j-i>l)
                {                          /* 计算一次消去所得新控制顶点 */
                        alfi=(u-U[i])/(U[i+ord+l]-U[i]);
                        alfj=(u-U[j-l])/(U[j+ord]-U[j-l]);
                        tempx[ii]=(Pwx[i]-(1.0-alfi)*tempx[ii-1])/alfi;
tempy[ii]=(Pwy[i]-(1.0-alfi)*tempy[ii-1])/alfi;  tempz[ii]=(Pwz[i]-(1.0-alfi)*tempz[ii-1])/alfi;
                        tempx[jj]=(Pwx[j]-alfj*tempx[jj+1])/(1.0-alfj);
tempy[jj]=(Pwy[j]-alfj*tempy[jj+1])/(1.0-alfj);  tempz[jj]=(Pwz[j]-alfj*tempz[jj+1])/(1.0-alfj);
                        i=i+1;   ii=ii+1;
                        j=j-1;   jj=jj-1;
                }                          /* while循环结束 */
                if(j-i<l)                  /* 检查节点可否消去 */
                {

if(Distance(int(tempx[ii-1]),int(tempy[ii-1]),int(tempz[ii-1]),int(tempx[jj+1]),int(tempy[jj+1]),int(temp
z[jj+1])) <= TOL) remflag=true;
                }
                else
                {
                        alfi=(u-U[i])/(U[i+ord+l]-U[i]);

if(Distance(int(Pwx[i]),int(Pwy[i]),int(Pwz[i]),int(alfi*tempx[ii+l+1]+(1.0-alfi)*tempx[ii-1]),
                                    int(alfi*tempy[ii+l+1]+(1.0-alfi)*tempy[ii-1]),
int(alfi*tempz[ii+l+1]+(1.0-alfi)*tempz[ii-1])) <= TOL) remflag=true;
                }
                if(remflag==false)      /* 不能再消去节点 */
                        break;              /* 跳出for循环 */
                else
                {                          /* 成功消去，保存新控制顶点 */
                        i=first;  j=last;
                        while (j-i>l)
                        {
                                Pwx[i]=tempx[i-off];    Pwy[i]=tempy[i-off];    Pwz[i]=tempz[i-off];
                                Pwx[j]=tempx[j-off];    Pwy[j]=tempy[j-off];    Pwz[j]=tempz[j-off];
                                i=i+1;   j=j-1;
                        }
                }
                first=first-1;   last=last+1;
        }                              /* 结束for循环 */
        if(l==0) return;
        int kk;
        for(kk=s+1; kk<=m; kk++)
                U[kk-1]=U[kk];           /* 节点移位 */
        j=fout;  i=j;
        for(kk=1; kk<l; kk++)        /* 重写pj到pi */
                if(fmod(double(kk),2.)==1.) i=i+1; /* 取kk除以2的余数 */
                else j=j-1;
        for(kk=i+1; kk<=n; kk++)     /* 顶点移位 */
        {
                Pwx[j]=Pwx[kk];   Pwy[j]=Pwy[kk];   Pwz[j]=Pwz[kk];   j=j+1;
        }
        return;
}
```